TEXAS TECH UNIVERSITY
Department of Computer Science™

Gnutella Style Peer-to-Peer (P2P) File Sharing

A JAVA Application (Test File)

by

Abhishek Kumar

Table of Contents

1. GitHub Link

Given below is the GitHub link of the Gnutella style peer-to-peer (P2P) file sharing system.

https://github.com/abhishek-kumar-code/Gnutella_Style_Peer-to-Peer-P2P-_FileSharing/tree/master/Gnutella

2. Maven Build - Building the project Gnutella using Star Topology

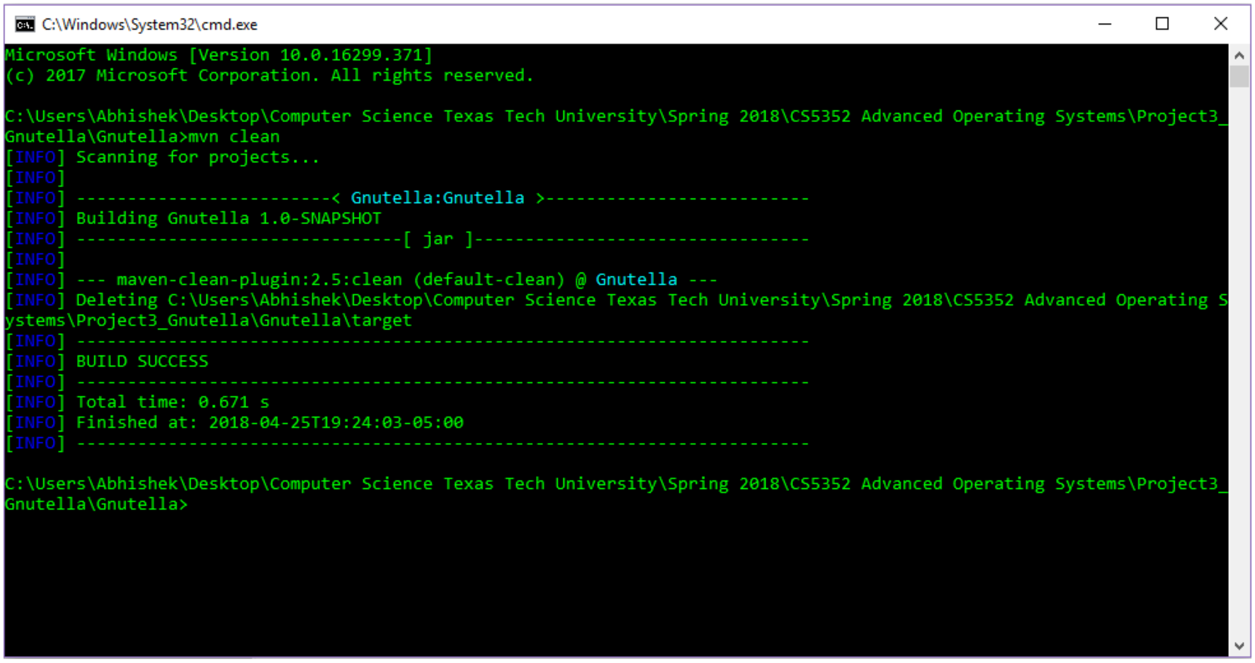
The following document explains how to build the project and execute the test cases.

The build tool used is **Maven**.

Instructions to build and run the Gnutella style peer-to-peer (P2P) file sharing system is discussed below.

Main

- Open command prompt and go to Gnutella folder that contains pom (XML document). The path file for my machine is given below: *C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_Gnutella\Gnutella*
- In the command prompt we type the command **mvn clean**. The Maven Clean Plugin, as the name implies, attempts to clean the files and directories generated by Maven during its build. This is shown in the figure below.

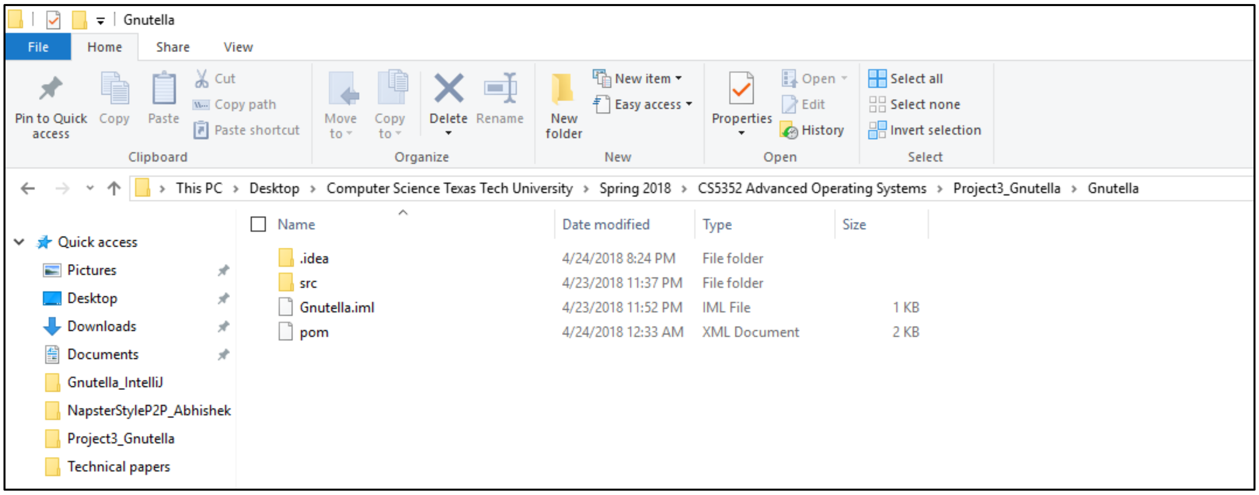


```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< Gnutella:Gnutella >-------------------------
[INFO] Building Gnutella 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ Gnutella ---
[INFO] Deleting C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating S
ystems\Project3_Gnutella\Gnutella\target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.671 s
[INFO] Finished at: 2018-04-25T19:24:03-05:00
[INFO] ------------------------------------------------------------------------

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella>
```

Fig 1: Output from the command mvn clean

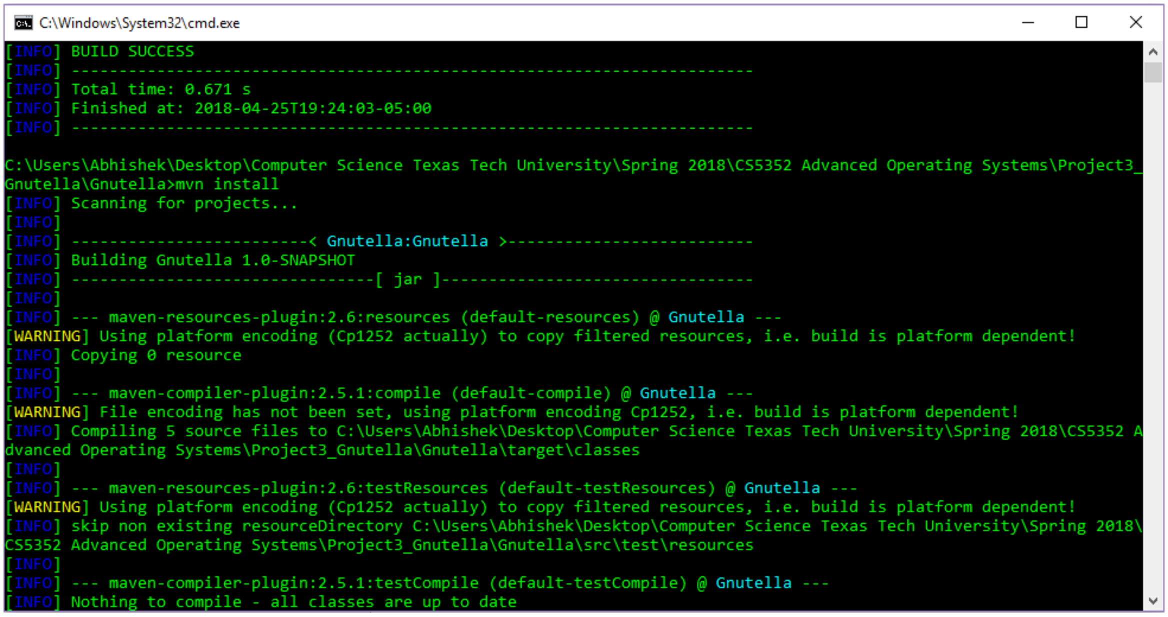


This PC › Desktop › Computer Science Texas Tech University › Spring 2018 › CS5352 Advanced Operating Systems › Project3_Gnutella › Gnutella

Name	Date modified	Type	Size
.idea	4/24/2018 8:24 PM	File folder	
src	4/23/2018 11:37 PM	File folder	
Gnutella.iml	4/23/2018 11:52 PM	IML File	1 KB
pom	4/24/2018 12:33 AM	XML Document	2 KB

Fig 2: Gnutella folder after running mvn clean

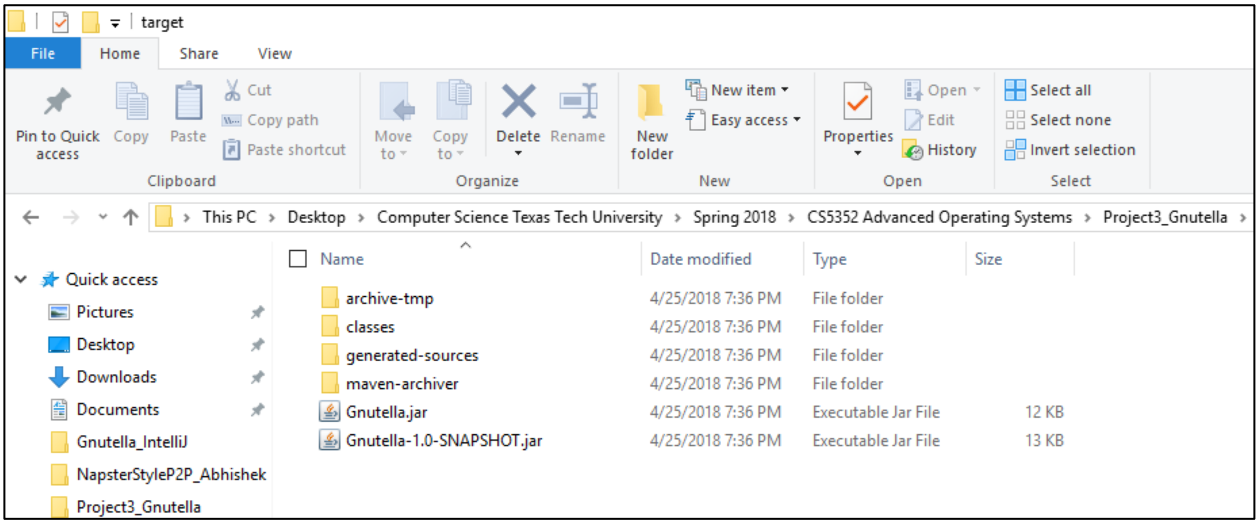
- Now, in the command prompt we type the command **mvn install**. This command installs the package into the local repository, for use as a dependency in other projects locally. We can clearly see that the Gnutella folder now has a target folder which contains the jar file. This format is common for distributing programs and libraries that are written in Java. Our .java files is compiled by the JVM to .class files.



```
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.671 s
[INFO] Finished at: 2018-04-25T19:24:03-05:00
[INFO] ------------------------------------------------------------------------

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella>mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< Gnutella:Gnutella >-------------------------
[INFO] Building Gnutella 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Gnutella ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ Gnutella ---
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 5 source files to C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 A
dvanced Operating Systems\Project3_Gnutella\Gnutella\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Gnutella ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\
CS5352 Advanced Operating Systems\Project3_Gnutella\Gnutella\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ Gnutella ---
[INFO] Nothing to compile - all classes are up to date
```

Fig 3: Output from the command mvn install

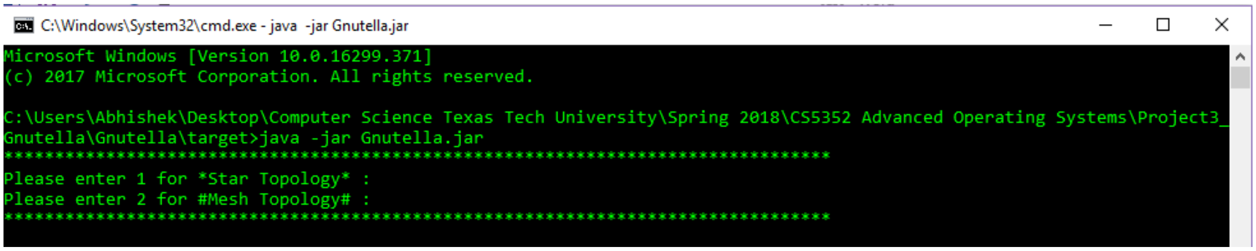


This PC › Desktop › Computer Science Texas Tech University › Spring 2018 › CS5352 Advanced Operating Systems › Project3_Gnutella ›

Name	Date modified	Type	Size
archive-tmp	4/25/2018 7:36 PM	File folder	
classes	4/25/2018 7:36 PM	File folder	
generated-sources	4/25/2018 7:36 PM	File folder	
maven-archiver	4/25/2018 7:36 PM	File folder	
Gnutella.jar	4/25/2018 7:36 PM	Executable Jar File	12 KB
Gnutella-1.0-SNAPSHOT.jar	4/25/2018 7:36 PM	Executable Jar File	13 KB

Fig 4: Target folder after running mvn clean

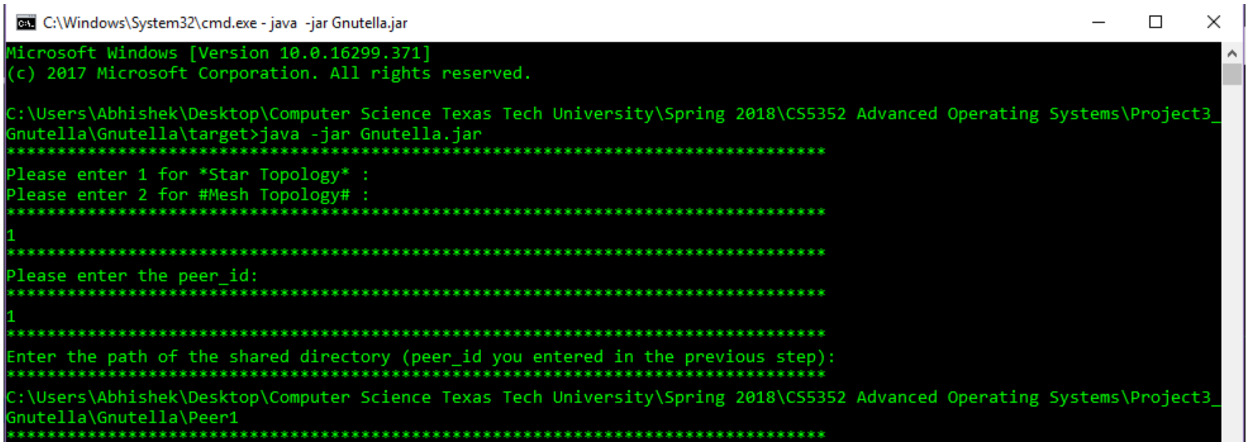
- **IMPORTANT**: Copy the **config.properties** file and the **mesh.properties** file from the Gnutella folder into the newly created target folder that contains the executable jar file.
- Open a new command prompt and go to target folder (under Gnutella folder) that contains the executable jar file. The path file for my machine is given below: *C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_Gnutella\Gnutella\target*
- In the command prompt we type the command **java –jar Gnutella.jar**. This runs the project as shown below. Follow the instructions as shown in the Fig 6 to run the project.



```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella\target>java -jar Gnutella.jar
*********************************************************************************
Please enter 1 for *Star Topology* :
Please enter 2 for #Mesh Topology# :
*********************************************************************************
```

Fig 5: Output from the command java –jar Gnuetella.jar

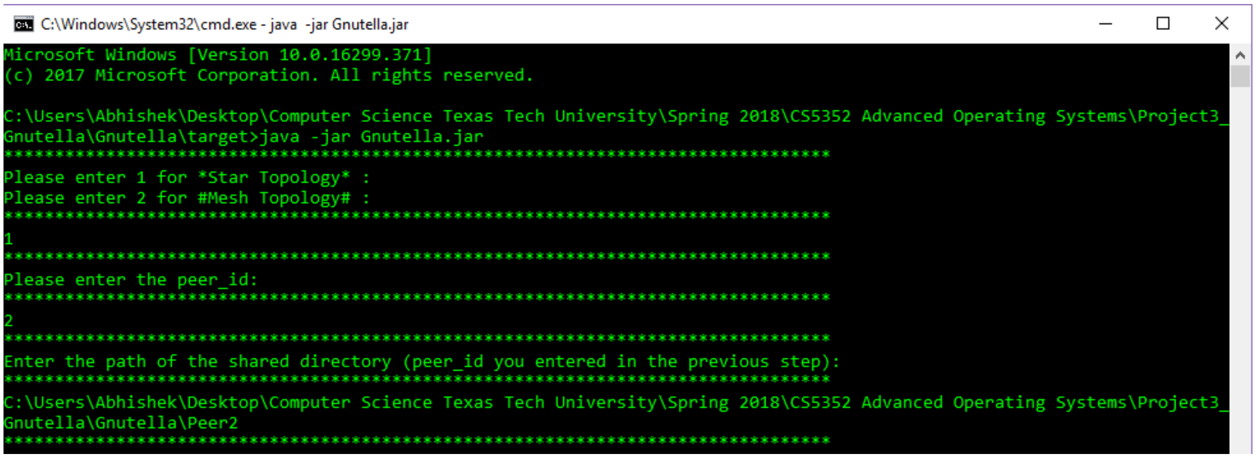


```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella\target>java -jar Gnutella.jar
*********************************************************************************
Please enter 1 for *Star Topology* :
Please enter 2 for #Mesh Topology# :
*********************************************************************************
1
*********************************************************************************
Please enter the peer_id:
*********************************************************************************
1
*********************************************************************************
Enter the path of the shared directory (peer_id you entered in the previous step):
*********************************************************************************
C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella\Peer1
*********************************************************************************
```

Fig 6: Star Topology distributed system file sharing (connecting peer#1)

- Now open a new command prompt and go to target folder (under Gnutella folder) that contains the executable jar file. The path file for my machine is given below: *C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_Gnutella\Gnutella\target*
- In this new command prompt we type the command **java –jar Gnutella.jar**. This runs the project as shown below. Here we run the same topology as show in Fig 6 with a different peer_id (say peer#2)

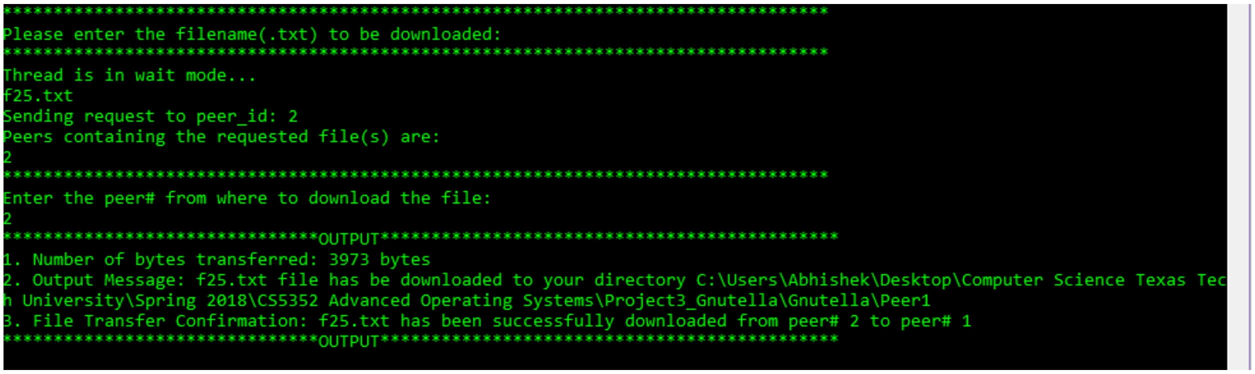

C:\Windows\System32\cmd.exe - java -jar Gnutella.jar

```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella\target>java -jar Gnutella.jar
*********************************************************************************
Please enter 1 for *Star Topology* :
Please enter 2 for #Mesh Topology# :
*********************************************************************************
1
*********************************************************************************
Please enter the peer_id:
*********************************************************************************
2
*********************************************************************************
Enter the path of the shared directory (peer_id you entered in the previous step):
*********************************************************************************
C:\Users\Abhishek\Desktop\Computer Science Texas Tech University\Spring 2018\CS5352 Advanced Operating Systems\Project3_
Gnutella\Gnutella\Peer2
*********************************************************************************
```

Fig 7: Star Topology distributed system file sharing (connecting peer#2)

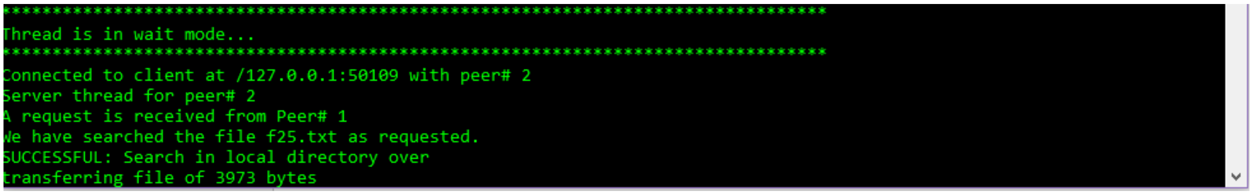
- Now go back to the previous command prompt (as shown in Fig 6) and enter the file that needs to be searched and downloaded to peer#1.



```
*********************************************************************************
Please enter the filename(.txt) to be downloaded:
*********************************************************************************
Thread is in wait mode...
f25.txt
Sending request to peer_id: 2
Peers containing the requested file(s) are:
2
*********************************************************************************
Enter the peer# from where to download the file:
2
*******************************OUTPUT*********************************************
1. Number of bytes transferred: 3973 bytes
2. Output Message: f25.txt file has be downloaded to your directory C:\Users\Abhishek\Desktop\Computer Science Texas Tec
h University\Spring 2018\CS5352 Advanced Operating Systems\Project3_Gnutella\Gnutella\Peer1
3. File Transfer Confirmation: f25.txt has been successfully downloaded from peer# 2 to peer# 1
*******************************OUTPUT*********************************************
```

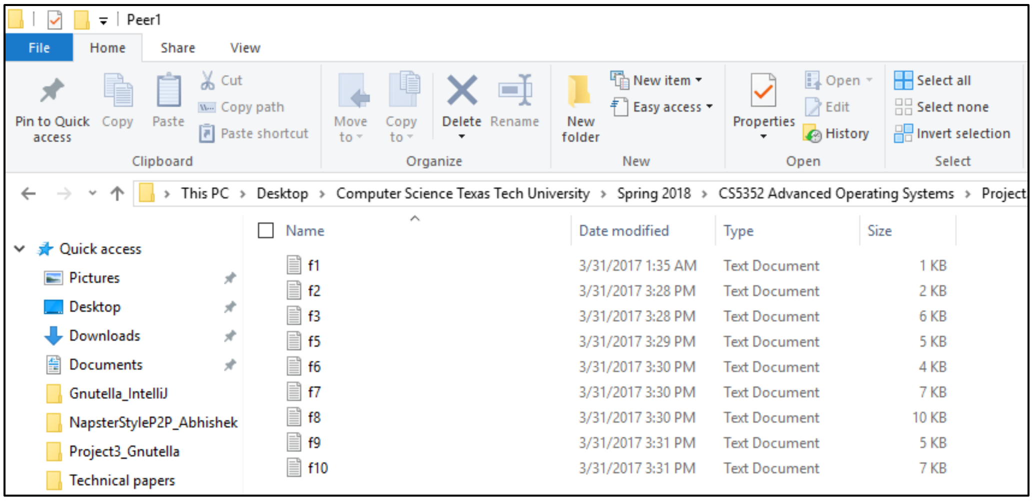
Fig 8: Star Topology P2P file transfer in distributed system (command prompt with peer#1)

As shown in Fig 9, the file of size 3973 bytes is transferred to the folder Peer1.



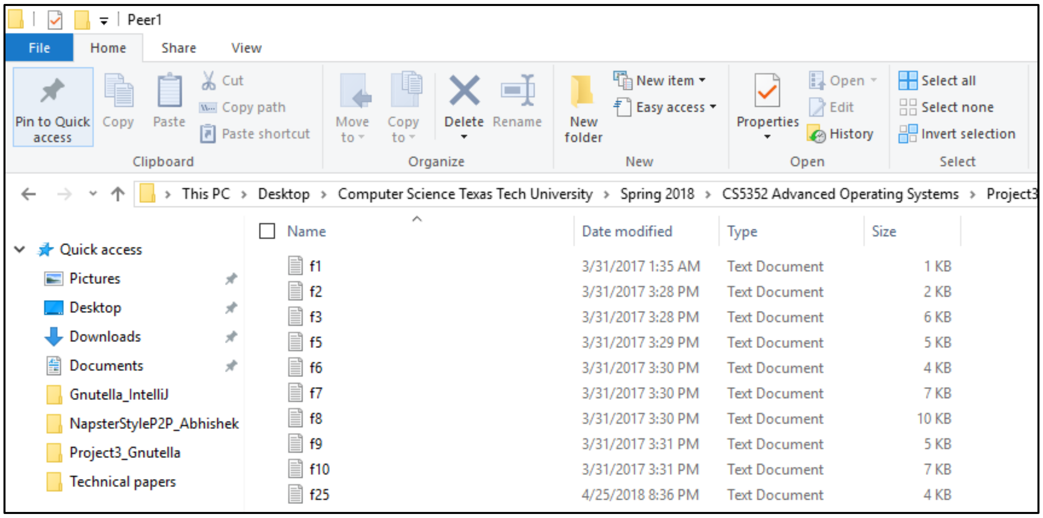
```
*********************************************************************************
Thread is in wait mode...
*********************************************************************************
Connected to client at /127.0.0.1:50109 with peer# 2
Server thread for peer# 2
A request is received from Peer# 1
We have searched the file f25.txt as requested.
SUCCESSFUL: Search in local directory over
transferring file of 3973 bytes
```

Fig 9: Star Topology P2P file transfer in distributed system (command prompt with peer#2)



Name	Date modified	Type	Size
f1	3/31/2017 1:35 AM	Text Document	1 KB
f2	3/31/2017 3:28 PM	Text Document	2 KB
f3	3/31/2017 3:28 PM	Text Document	6 KB
f5	3/31/2017 3:29 PM	Text Document	5 KB
f6	3/31/2017 3:30 PM	Text Document	4 KB
f7	3/31/2017 3:30 PM	Text Document	7 KB
f8	3/31/2017 3:30 PM	Text Document	10 KB
f9	3/31/2017 3:31 PM	Text Document	5 KB
f10	3/31/2017 3:31 PM	Text Document	7 KB

Fig 10: Peer 1 folder before file search and download



Name	Date modified	Type	Size
f1	3/31/2017 1:35 AM	Text Document	1 KB
f2	3/31/2017 3:28 PM	Text Document	2 KB
f3	3/31/2017 3:28 PM	Text Document	6 KB
f5	3/31/2017 3:29 PM	Text Document	5 KB
f6	3/31/2017 3:30 PM	Text Document	4 KB
f7	3/31/2017 3:30 PM	Text Document	7 KB
f8	3/31/2017 3:30 PM	Text Document	10 KB
f9	3/31/2017 3:31 PM	Text Document	5 KB
f10	3/31/2017 3:31 PM	Text Document	7 KB
f25	4/25/2018 8:36 PM	Text Document	4 KB

Fig 12: Peer 2 folder before file search and download

3. Maven Build - Building the project Gnutella using Mesh Topology

The following execution was performed on star topology. Using the similar instructions we can select option#2 if we want to perform mesh topology. The mesh topology will parse the mesh.properties file instead of config.properties file. This can be seen in the main.java code where it enters the else condition when choice given by user is 2. You can see the details of the mesh.properties file in the design documentation section (section 3, page#6, 7). The figure below depicts which peers communicate among each other in our distributed system.

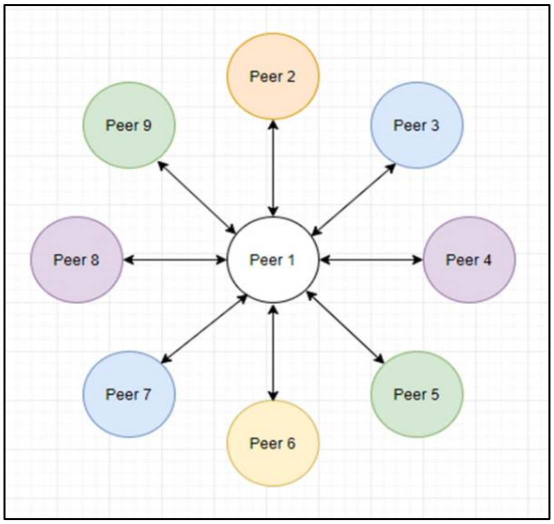


Fig 12: Star Topology

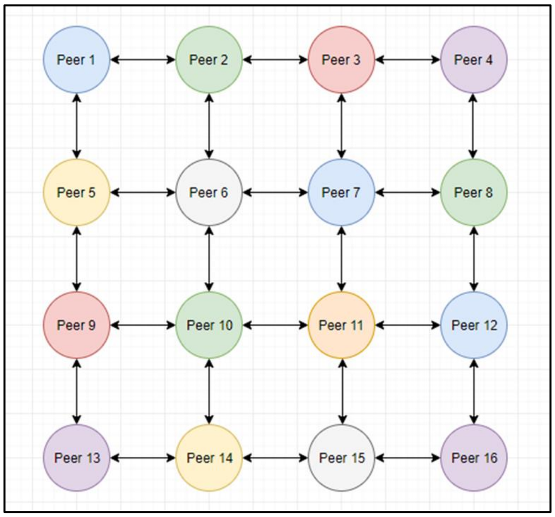


Fig 12: Mesh Topology